Seq-to-NSeq model for multi-summary generation

Guillaume Le Berre, Christophe Cerisara

Seq-to-NSeq model for multi-summary generation

Guillaume Le Berre[1] and Christophe Cerisara[2]

1- University of Lorraine - LORIA - France
guillaume.le-berre@loria.fr

2- University of Lorraine - LORIA - France
Address of Second Author's school - Country of Second Author's school

Abstract. Summaries of texts and documents written by people present a high variability, depending on the information they want to focus on and their writing style. Despite recent progress in generative models and controllable text generation, automatic summarization systems are still relatively limited in their capacity to both generate various types of summaries and capture this variability from a corpus. We propose to address this challenge with a multi-decoder model for abstractive sentence summarization that generates several summaries from a single input text. This model is an extension of a sequence-to-sequence model in which multiple concurrent decoders with shared attention and embeddings are trained to generate different summaries that capture the variability of styles present in the corpus. The full model is trained jointly with an Expectation-Maximization algorithm. A first qualitative analysis of the resulting decoders reveals clusters that tend to be consistent with respect to a given style, e.g., passive vs. active voice. The code and experimental setup are released as open source.

1 Introduction

Two main types of approaches are commonly used to automatically summarize text: extractive summarization generates summaries using a subset of the words from the original sentence while abstractive summarization rewrites parts of the original text with new words and syntactic structures. Abstractive summarization more closely mimics the way people actually write summaries, with paraphrasing and by exploiting the richness of natural language. Hence, in a summary, only a few proper nouns are entirely determined by the original sentence, while the rest of the words, as well as the chosen syntactic structures, strongly depend on the writing style of the person who writes the summary. Traditional sequence-to-sequence systems fail at capturing this variability and typically produce a unique summary that minimizes the average loss over all possible writing styles in the training corpus.

A popular way to bring variability in the system outputs consists in sampling from intermediate representations modelled as probability distributions, such as with Variational Auto-Encoders (VAE) [1]. However, random sampling does not enable to control the properties of the resulting summaries and our experiments show that, in practice, vanilla VAEs generate only a few real variations and that most differences concern only a few words.

We propose a different approach to capture and reproduce this variability, by enabling our network to directly output multiple summaries simultaneously. Intuitively, our model is based on a standard sequence-to-sequence network with attention and multiple concurrent recurrent decoders that are controlled by a discrete random variable. The whole model is trained using the Expectation-Maximization (EM) algorithm so that each decoder gets specialized by capturing a different writing style from the training corpus.

2 Related work

Sentence compression/summarization is a core task in Natural Language Processing (NLP) related to headline generation. Some previous works focused on syntactic structure and rewriting rules as in [2] or statistical machine translation techniques [3].

[4] introduced a new dataset for sentence compression extracted from the Annotated English Gigaword [5] They further proposed a model composed of a convolutional encoder and an attentive language model decoder inspired by [6]. Later works by [7] showed that replacing the basic language model by a Recurrent Neural Network improves the performance of the network. [8] further improved the model, finalizing its transformation into a full sequence-to-sequence network with LSTM encoder and decoder. Several additional improvements of these models have been proposed: among others, [9] exploited Minimal Risk Training and [10] proposed a selective encoding mechanism to control the flow of information sent to the decoder. Recently, [11] introduced a fully convolutional model with attention that obtains good performances for sentence summarization. [12] also proposed an EM-based generative model, closely related to our proposal. However, their algorithm uses an explicit clustering, while our model implicitly clusters the training data one example after another.

3 Our models

3.1 Baseline

Our baseline model is a sequence-to-sequence model with attention mechanism. Each word w_t of the input sequence $(w_t)_{1\leq t\leq T}$ is fed to an embedding layer $x_t = \text{emb}(w_t)$ and then into a bidirectional LSTM encoder [13] that produces a sequence of hidden states $(h_j^{\text{enc}})_{1\leq j\leq T}$.

The last state of this sequence h_T^{enc} is given as the initial hidden state to a second unidirectional LSTM decoder, which produces a new sequence of hidden states $(h_j^{\text{dec}})_{1\leq j\leq \tau}$ along with a summary $(\hat{w}_j)_{1\leq j\leq \tau}$.

During training, the j^{th} input to the decoder is the previous target word (teacher forcing) $x_j^{\text{dec}} = \text{emb}(y_{j-1})$. At test time, it is the previous generated word $x_j^{\text{dec}} = \text{emb}(\hat{w}_{j-1})$, and the length of the summary τ is determined at runtime when the special token $\hat{w}_j = \text{EndOfS}$ is generated. At each decoding step j, a fixed size attention vector c_j is computed as follow [14]:

$$a_j(i) = \frac{\exp(\text{score}(h_i^{\text{enc}}, h_j^{\text{dec}}))}{\sum_i \exp(\text{score}(h_i^{\text{enc}}, h_j^{\text{dec}})} \tag{1}$$

$$\text{score}(h_i^{\text{enc}}, h_j^{\text{dec}}) = (h_i^{\text{enc}})^T \cdot h_j^{\text{dec}} \tag{2}$$

$$c_j = \sum_{i=1}^{T} a_j(i) \times h_i^{\text{enc}} \tag{3}$$

Then the attention vector c_j is concatenated to h_j^{dec} and fed into a linear layer to output $\hat{w}_j$.

3.2 Proposed seq-to-Nseq model

The proposed model improves this baseline by generating diverse summaries with multiple concurrent decoders:

$$\hat{w}^1_{1\ldots\tau_1} = \text{dec}_1(h_T^{\text{enc}}) \quad \cdots \quad \hat{w}^n_{1\ldots\tau_n} = \text{dec}_n(h_T^{\text{enc}})$$

The model can be seen as multiple parallel sequence-to-sequence models with shared weights for the encoder, embedding and attention. Joint training of the encoder and decoders is done with an adaptation of the Expectation-Maximization algorithm, where, during training, a single decoder is sampled with a latent random variable $z \sim \text{Categorical}(n)$. This corresponds to a hard version of the EM algorithm, where the posterior distribution of z is approximated by a Dirac. Hence, assuming all decoders equiprobable:

$$p(z|w_{1:T}, y_{1:\tau}) \propto p(y_{1:\tau}|z, w_{1:T})p(z|w_{1:T})$$

$$\hat{z} = \arg\max_z p(z|w_{1:T}, y_{1:\tau}) = \arg\min_z -\log p(y_{1:\tau}|z, w_{1:T}) \tag{4}$$

which is the negative log-likelihood loss computed at the output of each decoder during training. The algorithm thus iterates through:

- Expectation: Make a forward pass and select $\hat{z}$ as in Eq-4;
- Maximization: Backpropagate from decoder $\text{dec}_{\hat{z}}$ and update parameters.

After training, each decoder is specialized to generate a specific type of summary. The key aspect is that this criterion is not defined a priori, but is rather automatically chosen to be representative of the variety of styles that occur in the training corpus. When applied to real use cases, the types of summaries learnt may be analyzed (see Section 5.2), and personalized summaries may be generated by choosing a decoder according to user preferences. We implement our model with a basic sequence-to-sequence network but the method is applicable to more complex encoder-decoder architectures.

In order to increase the difference between the summaries, we also experimented with a penalty cost for summaries that are too close from one another:

$$penalty = \sum_{i \neq j} \max(\alpha - CE(\hat{w}^i_{1...\tau_i}, \hat{w}^j_{1...\tau_j}), 0)$$

where α is a hyper-parameter and CE() is the cross-entropy loss. This system is referred to as "PEN". Another approach tested to increase this difference, named "NRI", consists in initializing the decoders parameters from distant points in the parameter space. Hence, our NRI model is initialized by splitting the training sentences into two groups: the shortest and longest sentences, each of the two decoders being pretrained on one group.

4 Experiments

4.1 Dataset

These models are validated on the English Gigaword 5th edition, which contains news articles from different sources. We use the sentence summarization version described in [4] (Extracted from the annotated version of Gigaword proposed in [5]). This dataset is built by pairing the first sentence of the articles in Gigaword with their headlines. The pairs first sentence/headline are then used as sentence/summary pairs. The train set is composed of 3.8M pairs. We use the test set provided by [4] (2000 samples) and a development set of 2000 samples.

4.2 Evaluation

The evaluation metric is the standard ROUGE [15] widely used in text summarization. We report the results for ROUGE1 (unigrams overlap), ROUGE2 (bigrams overlap) and ROUGE-L (longest common substring overlap). In addition, since our objective is to increase the variability of the generated summaries, we also report the average sentence difference and the edit distance between summaries, and perform qualitative analysis.

4.3 Hyper-parameters & Training

Our encoder is a bidirectional LSTM network with 2 layers and 250 hidden dimensions (500 after concatenation). All decoders are 2 layers unidirectional LSTMs with hidden size 500. The $50,000$ most frequent words of the vocabulary are encoded into 500-dimensional embedding vectors. The model is trained with 15 epochs of stochastic gradient descent with an initial learning rate of 1 divided by 2 at every epoch after the 8^{th} one. The mini-batch size is 64.

5 Evaluations

5.1 Quantitative evaluation

Table 1 compares the ROUGE scores of our models with related works on the Gigaword test set. Each individual decoder of the seq-to-Nseq model performs

	R1 (F)	R2 (F)	RL (F)
ABS [4]	29.55	11.32	26.42
ABS+ [4]	29.76	11.88	26.96
RNN MLE [9]	32.67	15.23	30.56
RNN MRT [9]	36.54	16.59	33.44
ConvS2S [11]	35.88	17.48	33.29
Var. Enc-Dec (VED)	35.29	16.77	32.83
seq-to-seq (baseline)	35.04	16.47	32.59
seq-to-2seq - output 1	34.90	15.81	32.27
seq-to-2seq - output 2	34.80	15.59	32.18
seq-to-3seq - output 1	32.51	13.65	29.37
seq-to-3seq - output 2	34.47	15.35	31.80
seq-to-3seq - output 3	34.93	15.77	32.56

Table 1: ROUGE F-measures on the Gigaword corpus.

slightly worse than the baseline seq2seq, which is expected since each decoder in the seq-to-Nseq model is trained to capture only a fraction of the training corpus. We thus also report the ROUGE in Table 2 where an oracle chooses, for every sentence, the best solution among $N = 2, 3$ proposals. These proposals correspond either to all decoder outputs, or to successive outputs of the baseline system after retraining. Of course, such ROUGE scores are not comparable to the state-of-the-art on this corpus, but they show that the variability in writing styles is better covered with the seq-to-Nseq model than with the baseline.

	R1 (F)	R2 (F)	RL (F)
Baseline (2x training)	38.58	18.97	35.84
seq-to-2seq (best)	39.48	19.49	36.68
Baseline (3x training)	40.26	20.15	37.28
seq-to-3seq (best)	41.26	20.77	38.09

Table 2: ROUGE-1 (F), ROUGE-2 (F) and ROUGE-L (F) on Gigaword when an oracle chooses the best among 2 and 3 candidate outputs.

Table 3 reports the percentage of pairs of summaries that differ by at least one word (*diff*), and their average *edit distance*. As in Table 2, each element of a pair is obtained either with retraining of the baseline, resampling (Variational Encoder-Decoder (VED)) or is the output of one of our decoders. Although the VED can generate many summaries through re-sampling, our results show that the resulting variability is much smaller than when retraining the baseline model. Conversely, our model's decoders generate the most diverse summaries.

5.2 Qualitative analysis

The following two examples illustrate typical differences between the outputs of each decoder of our model:

	diff (%)	edit distance
Baseline (2x training)	79.86	15.97
VED (2x sampling)	16.15	2.95
seq-to-2seq	89.44	21.77
seq-to-2seq+PEN	94.11	24.34
seq-to-2seq+NRI	96.26	26.48

Table 3: Variability of the generated summaries: the larger, the more diverse.

Input: Police arrested five anti-nuclear protesters Thursday after they sought to disrupt loading of French antarctic research and supply vessel, a spokesman for the protesters said
Target: Protesters target French research ship
Baseline: Five arrested in anti-nuclear protest **seq-to-2seq Output 1:** Five arrested in anti-nuclear protest **seq-to-2seq Output 2:** French police arrest five anti-nuclear protesters

Input: The head of the Russian-installed government in the breakaway republic of Chechnya narrowly survived a bomb attack Monday, the third assassination attempt against top Russian officials in two months.
Target: Head of UNK Chechen government survives bomb attack
Baseline: Chechen government survives bomb attack **seq-to-2seq Output 1:** Chechen leader survives bomb attack **seq-to-2seq Output 2:** Third assassination attempt in Chechnya

The following three examples illustrate the differences in passive vs. active voice. In the majority of the cases, when such differences occur, each decoder choice is consistent, i.e., the same decoder generates active voice while the other focuses on passive. More generally, the assignment of each training sentence to one or the other decoder is relatively stable after epoch 13, since about 80% of training examples stay assigned to the same decoder between epochs.

Output 1: Chinese pro-democracy activist arrested in Shanghai **Output 2:** Chinese police arrest pro-democracy activist

Output 1: Sudanese convicted of drug trafficking beheaded **Output 2:** Saudi Arabia beheads Sudanese convicted of drug trafficking

Output 1: Chinese fishing boat hijacked off east Africa **Output 2:** pirates hijack Chinese fishing boat in Somalia

6 Conclusion

An extension of the standard seq-to-seq model with attention is proposed to generate more diverse summaries than the current state-of-the-art text summarization systems. Conversely to related works, the diversity in writing styles is neither defined nor controlled a priori, but is rather automatically extracted from the training corpus. Multiple decoders/writers are thus trained, each one specialized in its preferred style. Both quantitative and qualitative analysis of the generated summaries on the Gigaword corpus confirm that a greater diversity may be achieved thank to the proposed model.

References

[1] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[2] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation, 2003.

[3] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 318–325, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[4] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685, 2015.

[5] Courtney Napoles, Matthew R. Gormley, and Benjamin Van Durme. Annotated gigaword. In *AKBC-WEKEX@NAACL-HLT*, 2012.

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[7] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98. Association for Computational Linguistics, 2016.

[8] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016.

[9] Ayana, Shiqi Shen, Zhiyuan Liu, and Maosong Sun. Neural headline generation with minimum risk training. *CoRR*, abs/1604.01904, 2016.

[10] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. *CoRR*, abs/1704.07073, 2017.

[11] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[12] Ershad Banijamali, Ali Ghodsi, and Pascal Poupart. Generative mixture of networks. *CoRR*, abs/1702.03307, 2017.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8), November 1997.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[15] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. 2004.